Notification/Alert Binary Format(Version 0x03)(/)

Notification/Alert Binary Format from Camera to Server (/)

Format Changes

This version of the binary alert will incorporate a few major changes which may affect skeleton data interpretation. The objective is to reduce the bandwidth and storage requirement on the alert and skeleton data.

Proposed changes are as follows

- Change all location coordinates from integer values to fixed point fractional representations
  - Add width and height in the alert frame header
- Change all probability values from floating point to fixed point representations
  - Change all the probabilities to 8bit fixed point representation
  - Possibility of hiding the key point definitions for security reasons
- Bit packing on skeleton data with key point probabilities: 10 bits for X, Y coordinates and 4 bits for confidence probability
  - Use 16 bit integer for X Y coordinates and 8 bit for confidence probability
- Use key points instead of bone parts for rendering
  - Need this change
  - Group key point probability with x y coordinates

Disclaimer

Revision Note

Version	Date	Author	Description
3	23 November 2020	Don Zhang	New alert format with bit savings

For alert version 0x03, the alert binary structure is defined as below (0x03):

Each alert message starts with the following alert header information()

alert version flag ()	epoch time (epoch) consistent with the epoch time format as analysis data (epoch)	person ID ()	skeleton Confidence ()	Explicit Padding ()	Screen Width ()	Screen Height ()	Skeleton X location (X)	Skeleton Y location (Y)	event type ()	level ()	number of frame ()
32 bits	32 bits	**signed** 32 bits	fixed point 8 bits	3 bytes	unsigned 16 bits	unsigned 16 bits	16 bits	16 bits	8 bits	8bits	16 bits

Each alert message contains skeleton data for multiple frames. The number of frames is specified by "number of frame" field in the alert header information.(. “”.)

Skeleton data for each frame begins with the following frame level header information.().

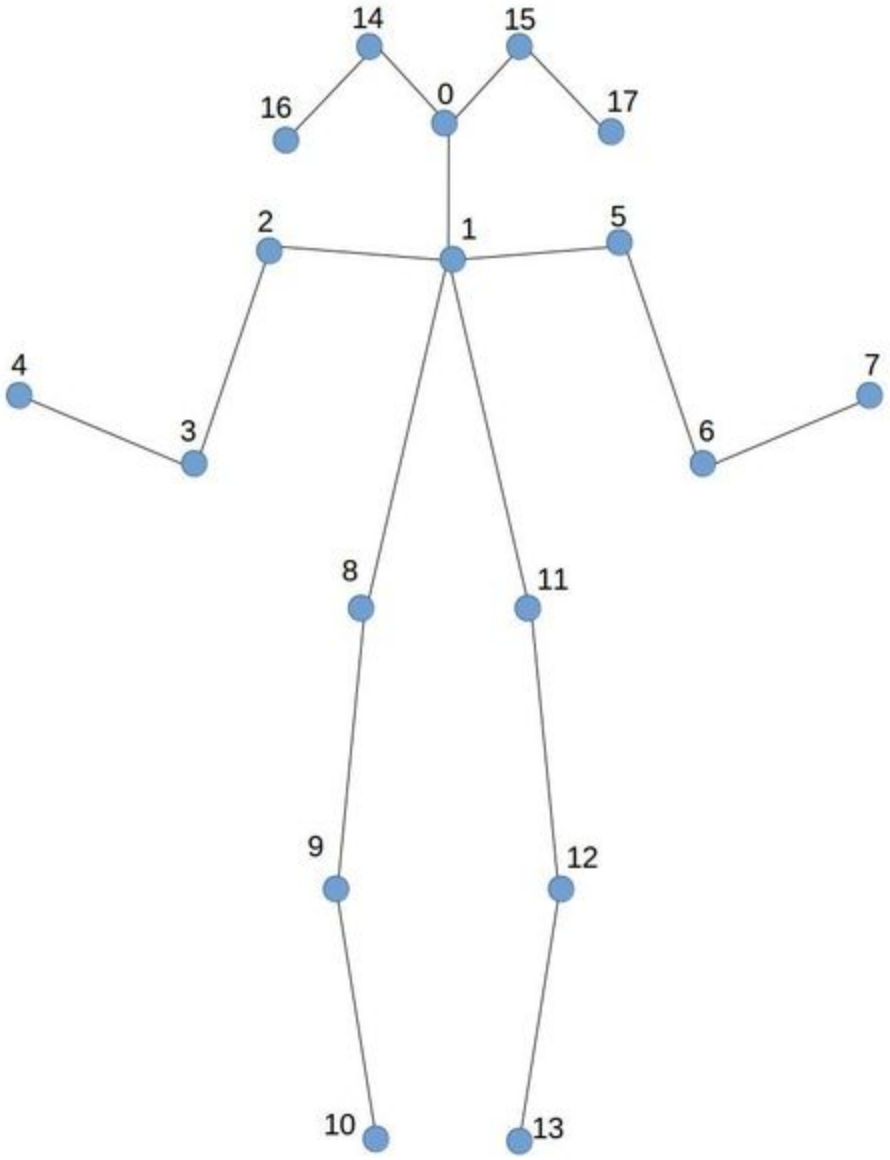
timestamp second(,) (delta from start in ms,)	action label ()	number of key points ()	human bounding box position structure ()	Action label probabilities ()
16 bits	8 bits	8 bits	8 bytes	8 bytes in total: 7 bytes, fixed point 8 bit representation plus 1 extra padding byte at the end

For each frame, there are a number of key point structures, specified by the field "number of key point structures" in the frame level header.(“”)

Each key point structure contains 2 signed 16 bit values for location and 1 fixed point probability. Specified by the following table.(2161)

Key Point Index ()	Key Point Probability ()	Key Point X (X)	Key Point Y (Y)
8 bits	fixed point 8 bits	**signed** 16 bits	**signed** 16 bits

Each complete human skeleton has 18 key points in total. Each key point has an ID, corresponding to the "Key Point Index" field in the key point structure. The "Key Point Index" are defined in the following diagram.(18ID“”“”)



C structures used to create the data(C):

```
struct BinaryAlertKeyPoint
{
    uint8_t keyPointIndex;
    uint8_t keyPointProb;
    int16_t X;
    int16_t Y;
};
```

```
struct cropBoxPos //human bounding box position structure
{
   int16_t x0;
   int16_t y0;
   int16_t x1;
   int16_t y1;
};
```

```
struct actionProbs
{
     uint8_t probs[7];
     uint8_t padding; //explicit padding
};
```

```
struct BinaryAlertFrameHeader

{

uint16_t msDelta;

uint8_t action;

uint8_t numKeyPoints;

struct cropBoxPos bbox;

struct actionProbs probAction;

struct BinaryAlertKeyPoint keypoints[0];

};
```

```
struct BinaryAlertHeader
{
uint32_t version;
uint32_t when; // unix epoch time
int32_t personID;
uint8_t skeletonConfidence;
uint8_t padding[3]; //explicit padding
uint16_t screenWidth;
uint16_t screenHeight;
int16_t x;
int16_t y;
uint8_t type;
uint8_t level;
int16_t numFrames;
struct BinaryAlertFrameHeader frames[0];
};
```